ACEEE

Proc. of Int. Conf. on Computational Intelligence and Information Technology

An Ant Colony Optimization based Advice Generation for Curriculum Sequencing under Flexible Learning System

Arpita Debnath[1], Paramita Sarkar[2], and Dr. Samir Roy[2]
[1] Dept. of Computer Sc. & Engg, NITTTR Kolkata, Block – FC, Sector – III, Kolkata – 106, India
Email: arpita.debnath03@gmail.com
[2] Dept. of Computer Sc. & Engg, NITTTR Kolkata, Block – FC, Sector – III, Kolkata – 106, India
Email: { hsarkarparamita.12, samir.cst }@gmail.com

***Abstract*— Curriculum sequencing aims to provide a customized optimal learning path through the curriculum landscape to individual learners since learners have different prior background knowledge, preferences, and goals. Researchers have given considerable attention to flexible curriculum sequencing control to provide adaptable, personalized learning programs. In an adaptive educational system, an optimal learning path aims to maximize a combination of parameters including the learner's understanding of courseware and the efficiency of learning the courseware. Flexibility in learning offers a learner an autonomy and control over his/her learning process as self-regulated learning. There exist a number of dimensions of flexible learning like class times, course content, location, organizational infrastructure, methods, learning styles, course requirements etc. This paper presents a scheme for advice generation, with the help of ACO, for curriculum sequencing under flexible learning system. The proposed frame work provides flexibility in time by asking the learner his affordability and allows his choice of the subjects to be learned by tracking his learning experiences. On the basis of these parameters, an optimized advice is generated, suggesting the learner, his future course of action, to be targeted towards the achievement of his learning objectives. The proposed system has been found to work satisfactorily with various test cases.**

***Index Terms*— Ant Colony Optimization, Flexible Learning, Curriculum Sequencing**

I. INTRODUCTION

Curriculum Sequencing is a technique used to build adaptive learning resources whose main objective "is to provide the students with the most suitable individually planned sequence of knowledge units to learn and sequence the learning tasks to work with" [1]. "..Flexible Learning is a set of educational philosophies and systems, that provides learners with increased choice, convenience, and personalization to suit them. Optimal curriculum sequencing helps the learners of heterogeneous categories according to their needs. In flexible educational system different learners can achieve the learning objectives in a self-paced manner. In a flexible learning environment, learners with different educational background and choices can have personalized curriculum planned for them. Shifting to flexible learning from traditional approach is challenging for the course planner since it has to be remembered that in this system, individual's need must be taken care of, considering his/her ability, availability and preferences. The main concern is how to plan for the new technique & learning environment and then to design a course that provides a requisite level of flexibility.

DOI: 03.LSCS.2013.6.144

II. PROPOSED WORK

The traditional teaching-learning processes promote a teacher-centric learning system which is rigid and lacks flexibility. Educational programmes for today's students should be specific to their needs i.e. student-centric rather than the former one. To meet the challenge, Universities provide a range of programmes delivered in a flexible mode. Further, the use of information technologies affects the teaching and learning process a lot. It is a challenging job for the course planners to how provide a requisite level of flexibility while shifting to the flexible learning mode form the traditional system. While designing the personalized curriculum for individual learners, certain constraints must be taken care of like – learners' prior background knowledge, availability, preferences. The proposed frame work provides flexibility in time by asking the learner his affordability and allows his choice of the subjects to be learned by tracking his learning experiences. On the basis of these parameters, an optimized advice is generated, suggesting the learner, his future course of action, to be targeted towards the achievement of his learning objectives.

III. PRELIMINARIES

A. Flexible Learning

1. Introduction to Flexible Learning

Flexibility is usually understood to mean offering students choices in their learning environment to better meet their individual needs. "..flexible teaching and learning is that mixture of educational philosophy, pedagogical strategies, delivery modalities and administrative structures which allows maximum choice for differences in student learning needs, styles and circumstances" [2]. In particular, flexible learning provides learners with choices about where, when, and how learning occurs. It is sometimes also referred to as personalized learning. Another frequently mentioned claim proponents make in many universities is the improved ability to correctly deal with student heterogeneity in learning preconditions such as pre-knowledge, motivation, or learning skills [3].

2. Dimensions of Flexible Learning

The diagram in Figure1 is presented as a visualization tool for describing the dimensions of flexible learning [15]. In flexible learning environment, the students are offered choices in time, space, methods, learning styles, contents, organizational infrastructure and entry requirements. Here, the key idea of flexibility is offering choices, though everything cannot be made flexible at all times for all students. Flexibility can be offered for more than one field. The dimensions for which flexibility is to be provided, depends on the requirements demanded by the system.

Figure 1. Dimensions of Flexible Learning

3. *Flexible & Self-Regulated Learning*

The characteristic of learner choices within learning processes is self-regulated learning (SRL). SRL is a process in which individuals take the initiative, with or without the help of others, in diagnosing their learning needs, formulating goals, identifying human and material resources, choosing and implementing appropriate learning strategies, and evaluating learning outcomes [4]. Therefore, SRL is a complex learning process that makes high demands on students for choices. Moreover, several studies and articles point out that SRL in particular makes high demands on learning skills. Paris and Paris pointed out that the phrase "self-regulated learning" "emphasizes autonomy and control by the individual who monitors, directs, and regulates actions toward goals of information acquisition, expanding expertise and self-improvement" [5]. Self-regulation corresponds with independently generated thinking, feeling, and connecting to the adaptation of personal objectives. Recent studies concerning flexible and self-regulated learning indicated that hypermedia and new information technology plays an important role in SRL. Other researchers also observed that online learning environments enable students to experience autonomy by allowing them to decide when, where, and what they can learn.

B. *Curriculum Sequencing*

1. *Introduction to Curriculum Sequencing*

CS technology exists to provide the student with the most suitable individually planned sequence of learning tasks to work with. It is becoming more and more popular for the learner who prefers to have a self-guided learning system. Thus, the objective of the CS is to replace the rigid, general, and one-size-fits-all course structure set with a more flexible and personalized learning path, considering both, the learner-related features as well as the curriculum-related constraints. These include the learner's background and prerequisites satisfaction as well as his/her motivation, level of performance, location and learning capabilities and style [6]. For a specific-learner, the requirements change as his/her knowledge improves due to learning. It is worth noting that a solution to the CS problem does not only assist students in finding the most efficient and suitable learning path, but also helps instructors to fine-tune their course structure and content, and discover areas of improvement, enhancing the whole learning / teaching experience. An optimal path selection algorithm for knowledge learning amongst multiple curricula has been developed for improving the efficiency of computer supported collaborative learning [13].

CS is an NP-hard problem [7]. Its complexity can be illustrated by an example [8] that demonstrates course sequencing for a Master in Engineering program containing only 23 courses. There are various constraints like prerequisite relations, fixed-order sequence for some itinerary courses, etc. Thus, a feasible sequence consists of the 23 courses arranged in a way satisfying all constraints. In this case, the total number of possible (valid and invalid) sequences (permutations) approaches 23!. The solution space becomes larger if sub-sequences are to be considered. A complete enumeration of all the possible sequences is clearly not feasible. This problem becomes even harder with a solution space that is much larger in a realistic situation where a learner's background knowledge, his/her learning style and similar student-related factors are considered. As the CS problem is NP-hard, heuristics and meta-heuristics are usually used to approximate its solutions, in particular Evolutionary Computation (EC) approaches.

2. *EC Approaches to solve CS problems*

Courses might contain several hundreds of subjects. So, for a course containing N subjects in the curriculum, we have to look for the optimal curriculum sequence that adapts to the pedagogical structure at first and then to student individual needs. The solution space of the CS problem comprises all the N! possible sequences or permutations. The solution space is even larger if sub-sequences are permitted. A feasible solution is the one that satisfies all the requirements mentioned above. CS is a problem with many parameters that need to be simultaneously optimized in a complex and changing learning environment. Approximate algorithms represent the only tractable way to tackle large instances of such a problem.

There are two distinct categories of EC approaches to the CS problem, depending on whether the solution incorporates experiences of other similar learners, called *Social Sequencing* [9], or it is based mainly on the individual learner, called *Individual Sequencing.* Social Sequencing abstracts away the individual properties of learners drawing efficient learning paths from the emergent and collective behaviour of a group of learners. The essence of these social systems stems from the well-established e-commerce recommendation systems, where services are marketed according to user interests based on a large amount of other similar customers' profiles. Swarm intelligence methods are widely used to solve such a problem. Individual

Sequencing focuses on the individual learner rather than the collective performance of learners. Individual Sequencing approaches base their decision on those properties and tag on a pedagogical structure. The student model and the pedagogical concept relation are therefore fundamental to these systems. The most important EC methods used so far for Individual Sequencing are GAs, PSO etc. Particle swarm optimization (PSO) and genetic algorithm (GA) agents are designed, built and tested in real and simulated scenarios. Results show both approaches succeed in all test cases, and that they handle reasonably computational complexity inherent to this problem, but PSO approach outperforms GA [14].

C. Ant Colony Optimization

1. Introduction to ACO

Swarm intelligence is an approach for problem solving inspired by the behaviours of social insects and of other animals. The most researched and successful among them is Ant Colony Optimization technique (ACO). It was introduced in the early1990's. The inspiring source of ACO is the foraging behaviour of real ant colonies. Many ant species have trail-lying trail-following behaviour while foraging: individual ants deposit a chemical substance called pheromone as they move from a food source to their nest. The pheromone trails are the favourable paths that should be followed by other members of the colony.

2. The Origins & Inspirations of ACO

The first ACO algorithm was introduced in the early 1990's by Marco Dorigo and colleagues, inspired by the observation of real ant colonies [10]. Ants are social insects and they lived in colonies. Their foraging behaviour of finding shortest paths between food sources and their nests provides the inspiration for the algorithm. While searching for food, ants initially explore the area surrounding their nest in a random manner. And while moving, they leave a chemical called *pheromone* on the ground. Ants can smell pheromone which guides other ants to find the food source. They tend to choose their paths marked by strong pheromone concentrations. It has been shown that the indirect communication between the ants via pheromone trails – known as stigmergy [11] (which is an indirect form of communication) – enables them to find the shortest paths between their nest and food sources.

3. The Optimization Techniques

The model proposed by Deneubourg and co-workers for explaining the foraging behaviour of ants was the main source of inspiration for the development of ant colony optimization. In ACO, a number of artificial ants build solutions to the considered optimization problem at hand and exchange information on the quality of these solutions via a communication scheme that is reminiscent of the one adopted by real ants. Different ant colony optimization algorithms have been proposed. The original ant colony optimization algorithm is known as *Ant System* and was proposed in the early nineties [12]. Since then, a number of other ACO algorithms were introduced.

IV. PROBLEM FORMULATION

A. Problem Statement

In a flexible learning environment, a learner with options in course may get confused to decide the suitable subjects for him/her to have the optimality. In the curriculum, there exist certain constraints related to the subjects like – prerequisites, credits, and required time. Therefore, the system should sequence subjects as per prerequisite constraint satisfaction. The system should be flexible to sequence the curriculum as per learner's affordability of time & learning experiences. System must take care of the constraint of maximum semesters, the required credit defined for the accomplishment of the degree as well as the required credit for major/minor subjects defined for the accomplishment of degree. Considering all these aspects, the system should advice the learner a few feasible sequences those may be optimal or near to optimal solutions for the learner to achieve his/her learning objectives.

B. Curriculum as asymmetrical directed graph

Curriculum is represented as complete directed asymmetric graph where nodes represents subjects .The directed edge represents the sequence in which subjects are covered. While moving in the graph if a node is traversed then it becomes visited node. The cost of edge from subject A to B:

$Edge_{A\text{-}B}$ = (time required to complete subject B) / ((credit gained after completing subject B)*(weight age given to the subject B)).

Those subjects, on which other subjects are based, have to be covered earlier to satisfy prerequisite constraints. Any predefined rule in the initial phase of ACO to restrict the (ant) agents with the choice of one node over other, has not been used. So, to take care of this matter, the time of the prerequisite subject is added to the numerator of the edge, if the prerequisite subject has not been covered yet. This will increase the probability of selecting the uncovered prerequisite subjects over the dependent subjects. So the new formula for the edge cost A to B while C is the prerequisite subject of B and C hasn't been covered yet:

$Edge_{A\text{-}B}$= (time required to complete subject B + time required to complete subject C) / ((credit gained after completing subject B)*(weight age given to the subject)).

If the subject C has been already covered by the learner earlier then the cost of the edge becomes:
$Edge_{A\text{-}B}$= (time required to complete subject B) / ((credit gained after completing subject B) * (weight age given to the subject B)).

The pictorial representation of the curriculum graph with four subjects is depicted in Figure 2 and the corresponding edge costs are represented in Table I.

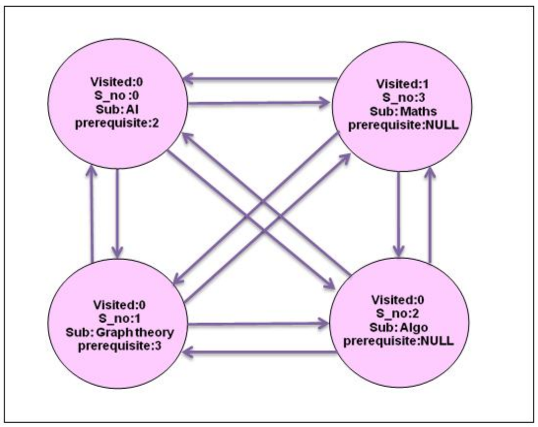

Figure 2. Curriculum as complete weighted asymmetric graph

TABLE I. CURRICULUM AS A COMPLETE WEIGHTED MATRIX REPRESENTING EDGE COSTS FROM FIGURE 2

[4]×[4]	S_no:0	S_ no:1	S_no:2	S_no:3
S_no:0	negative	x	y	z
S_no:1	u	Negative	v	w
S_no:2	q	r	negative	s
S_no:3	t	m	n	negative

x = (time required to complete subject (S_no:1))/ ((credit gained after completing subject (S_no:1))*(weight age given to the subject (S_no:1))).
y = (time required to complete subject (S_no:2))/ ((credit gained after completing subject (S_no:2))*(weight age given to the subject (S_no:2))).
z = (time required to complete subject (S_no:3))/ ((credit gained after completing subject (S_no:3))*(weight age given to the subject (S_no:3))).
u = ((time required to complete subject (S_no:0) + (time required by unvisited prerequisite subject (S_no:2)))/ ((credit gained after completing subject (S_no:0))*(weight age given to the subject (S_no:0))).
v = (time required to complete subject (S_no:2))/ ((credit gained after completing subject (S_no:2))*(weight age given to the subject (S_no:2))).
w = (time required to complete subject (S_no:3))/ ((credit gained after completing subject (S_no:3))*(weight age given to the subject (S_no:3))).
q = (time required to complete subject (S_no:0) + (time required by unvisited prerequisite subject (S_no:2)))/ ((credit gained after completing subject (S_no:0))*(weight age given to the subject (S_no:0))).
r = (time required to complete subject (S_no:1))/ ((credit gained after completing subject (S_no:1))*(weight age given to the subject (S_no:1))).
s = (time required to complete subject (S_no:3))/ ((credit gained after completing subject (S_no:3))*(weight age given to the subject (S_no:3))).
t = (time required to complete subject (S_no:0) + (time required by unvisited prerequisite subject (S_no:2)))/ ((credit gained after completing subject (S_no:0))*(weight age given to the subject (S_no:0))).
m = (time required to complete subject (S_no:1))/ ((credit gained after completing subject (S_no:1))*(weight age given to the subject (S_no:1))).

n = (time required to complete subject (S_no:2))/ ((credit gained after completing subject (S_no:2))*(weight age given to the subject (S_no:2))).

V. Methodology

A. Proposed Algorithm

The problem statement is to find the subject sequence satisfying the constraint of the prerequisite, the sequence must be ordered in such a way so that it should satisfy the learner's study hour constraint within the maximum semester constraint and the credit required to accomplish the degree (major credit lower bound and minor credit upper bound). It should also mould up with the learner's learning experience. The code have been modularized in three parts, in first part the method takes the input from the curriculum sheet and converts it into weighted edge cost matrix which represents the curriculum as a complete directed asymmetric graph. The output of weighted edge cost matrix used as input to the second method which is Ant colony optimization method which gives a solution set that has an optimal sequence of path or approximate to optimal sequence. This sequence is then fetched one by one considering the learner's requirements and is arranged in the semesters which are done by the third procedure of scheduling.

1. Implementation of procedure for formation of weighted graph matrix with curriculum

The implementation of the procedure for the formation of weighted matrix with the curriculum needs data from the subject database which gives information related to each subject and data from the learner's record which stores learner's previous detailed records. The subject database consist of unique Id for each subject, its name, its subject code which signifies whether the subject is major , minor or compulsory subject ,the credit corresponding to the subject and the time required to complete the subject. The student database consist of the details of the credit obtained till date, the semester covered till date, the major/minor/compulsory credit and the information about each subject whether covered or left. The learner must provide the time that she can afford to study per day. The learner is also required to suggest her weightage for the uncovered subject if she wants to customize her learning experience. If the learner doesn't want to customize the learning experience the system by itself will give weightage to the uncovered subjects and sequence the subject accordingly. The basic information generated with this procedure is to generate an indirect dependency between the dependent subjects with their prerequisite subject. So this procedure emphasizes on the issue that if prerequisite subject hasn't been covered yet then the selection probability of prerequisite subject should be more than the independent subject by the agent (ant).

➢ Data Structure for the procedure - A subject is represented by a vertex node of the graph and is represented as a structure. All the subjects are represented as array of the structure. The information from the learner's record i.e. the information about the subjects already learned by the student or is yet to be learned are stored in the array of flag bit corresponding to each subject. The learner's choice weightage is also stored in float array corresponding to each subject.

Representation of problem data:
Structure Subject[n] - %structure array of subject
Flag visited[n] - %subjects learned or not learned by the learner
Float choice_weightage[n] - %weightage given by the learner to unlearned subject
Double weight_matrix[n] [n] - %the weighted matrix evaluated from the graph as result

Structure of subject:
Begin
Integer Serial_No - %uniquely identifies a subject
Char Sub_name [20] - % subject name
Char sub_code - % code symbolizes subjects as major minor or compulsory subjects
Integer credit - % credit associated with the subject
Real Time - % time required to complete the subject
Integer Prerequisite - % the Serial_No of the prerequisite subject on which subject is dependent
End

➢ Algorithm for Procedure Weighted_matrix_curriculum - The main objective of the algorithm is evaluation of the weighted matrix with the curriculum graph on which ACO will further run.

Input: Learner have to enter maximum affordable hour that the learner can afford to study per day. Then the learner is asked whether she wants to customize her learning experience or not. If the learner is interested to customize her learning experience then the learner is asked to provide weightage against each unlearned subject with the system provided choices. If the learner doesn't want to customize then the system itself will give default weightage to all subjects.
Output: Weighted matrix of [n][n] order where n is the number of subjects present in the curriculum.

Initialize data

Structure Subject [n] array values are fetched from the database of the subject.
Visited[n] values are fetched from the learner's database file.
Choice_weightage[n] values are gained from either as input from the learner or from the system.

Weighted_matrix_curriculum (structure subject[n], flag visited[n], Float choice_weightage[n])

Procedure Weighted_matrix_curriculum
Begin
for each and every edge between vertex i to j - % here i, j and k are serial number of the corresponding subject nodes in the curriculum graph. As the graph contains no self loops so i shouldn't be equal to j.

if (subject$_i$ has a prerequisite subject$_k$ and that subject$_k$ hasn't been covered by the learner previously)

weight_matrix[i][j]=(time required to complete subject$_j$ + time required to complete subject$_k$)/
(Credit gained after covering subject$_j$ + weightage given by the learner to the subject$_j$)

end if
else

weight_matrix[i][j] = (time required to complete subject$_j$)/(credit gained after completing subject$_j$+ weightage given by the learner to the subject$_j$)

end else

end for
End procedure

The weight_matrix[i][j] signifies the cost evolved in covering subject j after covering subject i. This cost is evaluated with respect to time required to complete subject j. The credit gained by covering subject j and the weightage given by the learner for customizing her learning experience to the subject j. It also brings up the dependency indirectly prevailing between the subjects and its respective prerequisite subject. This is done by adding the time of the prerequisite subject to the dependent subject's time so increase the probability of the prerequisite subject to be opted first by the ant (agent) before selecting dependent subject. By adding the cost of the prerequisite subject's time to the dependent's subject time it has been tried to decrease the probability of its edge selection over prerequisite subject's edge selection ,as here ACO works on the minimization of edge trail cost.

2. Implementation of procedure ACO for optimized sequence of curriculum

The implementation of the procedure is for the generation of optimized sequence of subjects of a curriculum. This sequence is generated by taking the weighted graph of the curriculum. The first iteration of ACO each ant initiate from random vertex node and the search is simple and straight forward as it is greedy construction procedure of local solution set by each ant based on nearest neighbour edge cost as heuristic. After the first iteration onward pheromone trails are also initialized based on the trail traversed by each ant. After the first run the construction procedure for local solution set becomes greedy towards: a) Pheromone trails available in common memory space, b) Heuristic information based on previous solution. These two factors influence the search of ants to produce the global optimal solution constructively. The entire procedure is repeated till the global solution converges at a point.

➢ *Data Structure for the procedure* - The Ants are represented as array where the first element of a one dimensional array in [n][n+2] array will represent the cost evolved in traversing the trail by an ant and rest unit of the same one dimensional array will fetch the path trail traversed by the ant as these ants have no local memory. The number of ants is equal to number of nodes in a graph. The pheromone values are stored in the common memory region shared by the each ant and called as the pheromone matrix which is of dimension

[n][n].The Weighted_matrix is taken as input for the edge matrix for the ACO of dimension [n][n].The choice_info[n][n] matrix will store information which will influence further search of the local optimal solution set of the ant. The array of boolean values visit[n][n] will store the information for each ant whether the vertex node corresponding to the trail of each ant has yet been visited or not.

Representation of problem data:

Float Ant[n][n+2] - %Ant matrix storing ant's trail and its cost where is the number of ants= number of vertices in the graph.

Float Pheromone[n][n] - %Pheromone matrix stores pheromone value corresponding to each edge in the graph.

Double Edge[n][n] - %Edge matrix obtained from weighted_matrix[n][n] storing cost of edges of the graph.

Choice_info[n][n] - %Choice_info matrix influences further search of ant after initialization.

Boolean Visit[n][n] - %Visit matrix stores the information of the ant trail about which of the vertices have been visited by the corresponding ant in its trail.

Integer Soln_path[n][n] - %stores the best solution path gained from the ACO procedure

Representation of Ants :

Ants are represented as a matrix of dimension [n]×[n+2] where the first element of a one dimensional array of the two dimensional array represents the cost of path trail traversed by the respective ant signified by the ant number and rest of the elements of the one dimensional array of the two dimensional array represents the trail traversed by the ant. In the figure3, Ant [5]×[7] represents ant's trail cost and the trail traversed by the respective ant. Like for ant[0] Ant[0][0] represents trail while traversing the trail of A (source),E,B,D,C and A(source) . Similarly rest of the ants has their trail cost and trail sequence in their respective one dimensional array. Table II represents the Ant Matrix for Figure 3.

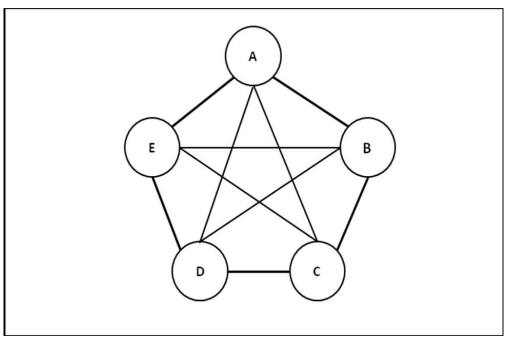

Figure 3.Graph for ant trail

TABLE II. REPRESENTATION OF [5]X[7] ANT MATRIX BASED ON FIGURE 3

[5]×[7]Ant	Cost of trail	Source	1	2	3	4	Source
Ant 0	A→E→B→D→C→A	A	E	B	D	C	A
Ant 1	B→D→E→A→C→B	B	D	E	A	C	B
Ant 2	E→D→B→A→C→E	C	D	B	E	A	C
Ant 3	D→B→E→A→C→D	D	B	E	A	C	D
Ant 4	C→D→B→E→A→C	E	D	B	A	C	E

➢ *Algorithm for Procedure ACO* - The main task of the algorithm is to produce best optimal or approximately optimal solution sequence of the trail evaluated by the ants within the graph.

Input: The input for this procedure is taken from the Weighted_matrix_curriculum( ) procedure which produces weighted_matix[n][n] which is used to initialized the Edge[n][n] matrix.

Output: The matrix soln_path matrix stores the best trail found by the ants.

Initialize data

The weighted_matrix[n][n] taken as to initialize Edge[n][n].

The pheromone[n][n] is initialized with all values as 0.

Choice_info[n][n] initialized to all 0 values.

The Visit[n][n] initialized to all values as 0 signifying all as unvisited vertices.

Procedure ACO( )

Begin

Step 1: Initialize data.

Step 2: Construct local solution set by each ant initialized from random selection of vertices based on greedy for 1/Edge cost heuristics

Step 3: Update the pheromone trail for local solution set.

Step 4: Choose best local trail as globally optimized solution
Step 5: Update Pheromone trail for global solution.
Step 6: Update Choice_info matrix.
Step 7: While(solution Converges) do
  Construct local solution set by each ant initialized from random selection of vertices based on Choice_info matrix for local solution set.
  Update Pheromone trail for local solution set.
  Choose best Local trail as globally optimized solution.
  Update Pheromone trail for global solution.
  Update Choice_info matrix.
 End while
Step 8: Choose the best trail gained from step 7 store it in sol_path[n][n] matrix.
End procedure

VI. RESULTS & DISCUSSION

In this proposed work, DEV C++ has been used as the implementation tool. Our work has implemented a system that asks for the registration no. from an existing registered learner & provides him/her the flexibility of time, how much he/she can devote per day on an average. It also provides the facility to customize the learning experience by asking the learner's interests on different subjects. After having all these information from the learner & fetching the present status of the learner from the database, an advice is generated. Here, a set of students has been analyzed with different devotion time and a result of that comparative study is shown in Figure 4. The more time can be spent, the less no. of semesters required to complete the course. It shows that a learner may opt for flexibility in time according to his/her requirements considering the constraint of total no. of permissible attempts. It's not always possible for learner to consider all the aspects simultaneously of a curriculum sequence so the system has provided a way out from such a quest of a learner.

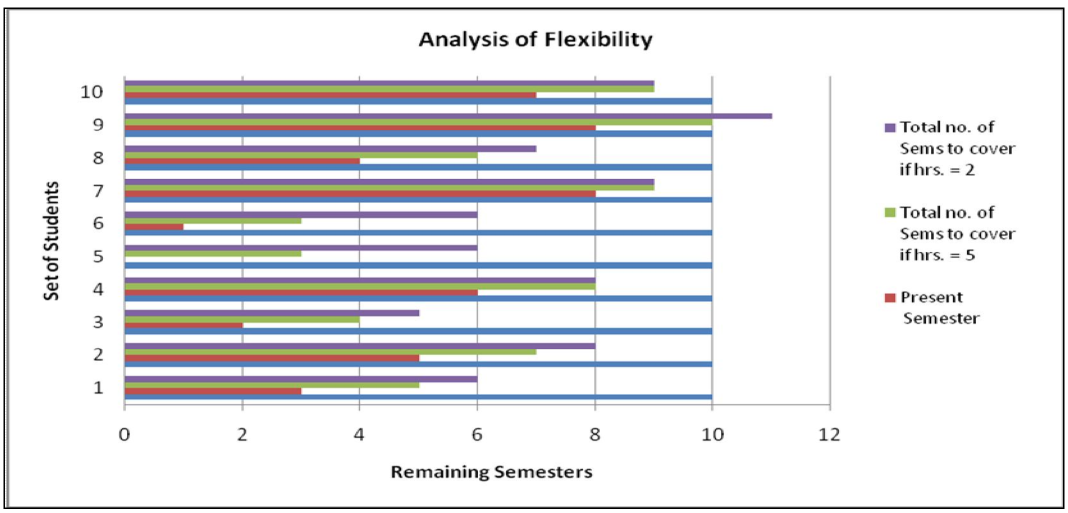

Figure 4.Comparison of flexibility of time for the learners

VII. FUTURE RESEARCH DIRECTION

Here the system is sequencing the curriculum in static sense when the learner is providing the weightages to the subjects once at the beginning. But the more stringent form of it appears when this problem takes a dynamic form. The dynamic form of this problem should be considered because it is always possible that the learners' desirability or preferences may change with the acquirement of knowledge gained from previously sequenced subjects in the curriculum. The same problem will become more complex if the sequencing is done with respect to the Learning Objects (LOs) present in a course. A learning object can have more prerequisites than a subject.

REFERENCES

[1] P. Brusilovsky, "Adaptive and Intelligent Technologies for Web-based Education," Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, vol. 4, pp. 19-25, 1999.

[2] Lundin. R. “Flexible Delivery: An International perspective”, UQ Teaching and Educational Development Institute In-House Conference, Initiatives in Flexible Delivery, September, 1997, http://www.tedi.uq.edu.au/tei/flex delivery/Lundin.html.

[3] Cornelius, S., & Gordon, C. (2008). Providing a flexible, learner-centred programme: Challenges for educators. The Internet and Higher Education, 11(1), 33-41. doi:10.1016/j.iheduc.2007.11.003.

[4] Knowles, M. (1975). Self-directed learning: A guide for learners and teachers. New York: Association Press.

[5] Paris, S. G., & Paris, A. H. (2001). Classroom applications of research on self- regulated learning. Educational Psychologist, 36(2), 89–101. doi: 10.1207/S15326985EP3602_4.

[6] Hauger D, Kock M (2007) State of the art of adaptivity in e-Learning platforms. Workshop at adaptivity and user modeling in interactive systems ABIS 2007, Halle/Salle, Germany. doi:10.1.1.91.6409.

[7] Acampora G, Gaeta M, Loia V (2009) Hierarchical optimization of personalized experiences for e-Learning systems through evolutionary models. Neural Comput Appl. doi:10.1007/s00521-009-0273-z.

[8] de-Marcos L, Martinez JJ, Gutie´rrez JA (2008a) Swarm intelligence in e-learning: a learning object sequencing agent based on competencies. In: Proceedings of the 10th annual conference on genetic and evolutionary computation. ACM Special Interest Group on Genetic and Evolutionary Computation, pp 17–24.

[9] Gutie´rrez S, Pardo A (2007) Sequencing in web-based education: approaches, standards and future trends. Studies in Computational Intelligence (SCI), vol 62. Springer-Verlag, Berlin, pp 83–117.

[10] Christian Blum, “Ant colony optimization: Introduction and recent trends”, in proc. Physics of life reviews 2 (science direct) pp.353-373, 2005.

[11] Marco Dorigo, Mauro Birattari, and Thomas St¨utzle, “Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique”, in proc. IRIDIA – TECHNICAL REPORT SERIES: TR/IRIDIA/2006-023.

[12] Marco Dorigo, Mauro Birattari, and Thomas St¨utzle, “Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique” in proc. IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE | NOVEMBER , pp. 28-39, 2006.

[13] Wenan Tan ; Guosheng Zhu ; Chuanqun Jiang ; Yi Du “An optimal path selection algorithm for knowledge learning amongst multiple curricula within computer supported collaborative learning ” in proc. International Conference on Computer Science and Service System (CSSS), 2011, pp. 1099-1101, Digital Object Identifier: 10.1109/CSSS.2011.5975056, 2011.

[14] de-Marcos, L. ; Martinez, J.J. ; Gutierrez, J.A. ; Barchino, R. ;Gutierrez, J.M. “A new sequencing method in Web-based education ” in proc. IEEE Congress on Evolutionary Computation, 2009. CEC '09, pp. 3219-3225, Digital Object Identifier: 10.1109/CEC.2009.4983352, 2009.

[15] Book: John Casey and Pam Wilson “A practical guide to providing flexible learning in further and higher education”, 2005.